Inspiring Excellence

CSE422: Artificial Intelligence

Project Name: Customer Category Classification

Group: 14

Section 21

Submitted By:

Md Tanvir Siam; ID 22299433

Nisat Nisa; ID 22299095

Submitted To:

Rafeed Rahman; Senior Lecturer

Department of CSE, Brac University

Riazul Islam Rifat; Lecturer

Department of CSE, Brac University

Introduction

We have executed a Customer Segmentation analysis using various Machine Learning algorithms. This project focuses on classifying customers into distinct segments based on lifestyle and earning source. The segmentation labels serve as the target classes, making this a classification problem. The dataset includes features such as age, annual income, work experience, family size and spending score. By training classification models on these features, we aim to predict the segment a customer is likely to belong to. After training multiple models, we compared their performance using visual evaluation metrics to identify the most effective approach for customer segmentation.

Dataset description

The dataset we were working on was a CSV file of 11 columns and 8068 rows. 11 features indicate the number of types of data given in the dataset and 8068 data points are here as the row count is from 0 to 8067. The Categorical features here are Gender, Ever_Married, Graduated, Profession, Spending_Score, Var_1, Segmentation columns. On the other Hand Quantitative features are Age, Work_Experience, Family_Size. ID is Not a useful feature for prediction but we will count it as numerical data. If a given dataset's targeted column values is object/string type and there are less than 20 types of values in that column then it's a Classification problem, else if the targeted column's values are numeric and float numbers with many distinct values it's a Regression problem. Target values are discrete integers with few classes and can be a Classification problem also (e.g., 0, 1, 2). Clearly this dataset is a Classification problem as its target column is Segment which has 4 unique values A, B, C and D. We ran a correlation test among all the numerical data and all the data again and found no correlations between data which indicates we can’t merge column or data type. To check

whether all unique classes in output feature ('Segmentation') have an equal number of instances wwe counted all four values and found out all instances have different occurrences like D have 2268, A have 1972, C have 1970 and B have 1858 occurrences. We also executed some operations to analyze the data. Operations like describing data, skewness finding, bar chart, sns heatmap and hist diagram are performed in code.

Dataset pre-processing

On inspecting the dataset using isnull().sum() we found that several columns contain missing (null) values. For ‘Ever_Married’ column 140, for ‘Graduated’ 78, for ‘Profession’ 124, for ‘Work_Experience’ 829, for ‘Family_Size’ 335 and for ‘Var_1’ 76 datasets are found null. For those missing values we used imputation using SimpleImputer with strategies like 'mean' or 'most_frequent' or row deletion when nulls were minimal and did not significantly affect the dataset size. Columns like 'Gender', 'Ever_Married', 'Graduated', 'Profession', 'Spending_Score', 'Var_1' and the target 'Segmentation' are categorical and needed to be converted into numerical values to perform correlation or apply ML models. We used LabelEncoder to convert categorical columns to numerical form. This allows models and correlation analysis to be applied properly. Numerical columns vary in scale. For instance, 'Age' ranges differently than 'Work_Experience' or 'Annual_Income'. Without scaling, models like KNN or Neural Networks can perform poorly. We used MinMaxScaler to scale the features to a uniform range between 0 and 1, ensuring models like KNN and Neural Network perform optimally.

Dataset Splitting

The dataset was split using the train_test_split function. Since this is a classification problem with balanced-enough class distribution, we applied random splitting with a 70:30 ratio as instructed. This ensures 70% data for training and 30% for evaluating model performance.

Model Training and Testing

We applied Logistic Regression, Naive Bayes and Neural Network models. Here, Logistic Regression is used for binary/multiclass classification problems, Naive Bayes is used for probabilistic classification and Neural Network is used for classification problems. Each model was trained using the .fit() function and evaluated on the test set using .predict().

Model selection/Comparison analysis

The accuracy of each model was compared using a bar chart. Neural Networks showed the highest accuracy of 51.47 followed by Logistic Regression of 49.48. For Naive Bayes the accuracy is 48.3. We used multiple metrics to compare models like Precision, Recall, F1-Score using classification_report(), Confusion Matrix using confusion_matrix()and ROC Curve and AUC Score using roc_curve() and auc(). These allowed us to visualize and understand model strengths and weaknesses.

Conclusion

From this customer segmentation project, we observed that Models performed reasonably well in classifying customer segments based on the given features. Neural Network achieved the best performance, likely due to its ability to learn complex feature interactions. Again, Naive Bayes was fast but less accurate due to its strong independence assumptions. Lastly, for Logistic Regression performed moderately well in comparison overall. Here, we faced several difficulties while processing and applying models in the following dataset. Like, handling missing and categorical values properly, ensuring proper scaling before using distance based models, interpreting results of multiclass ROC curves and confusion matrices. Overall, this project deepened our understanding of data preprocessing, model selection, and performance evaluation in a practical classification task.

Reference

Finalizing.ipynb